

# JAK SI SESTAVIT VLASTNÍHO ROBOTA

METODICKÉ LISTY



## MODELOVÁNÍ PODVOZKU

### ÚVOD

Základem každého robota je jeho podvozek. Ten je možné vyrobit z různých materiálů. Jeho jedinou podmínkou je, aby se na něj všechno vešlo. Žáci si sami v této sekci vymodelují podvozek v programu TinkerCad a tento model si poté vytisknou na 3D tiskárně.

#### **Cíl sekce:**

Žák vytvoří jednoduchý podvozek podle návodu v programu TinkerCad. Žák čte s porozuměním jednoduché technické nákresy. Žák pod vedením vyučujícího vytiskne podvozek na 3D tiskárně.

#### **Praktická poznámka:**

Program TinkerCad je online na adrese tinkercad.com. Pro používání programu je potřeba mít vytvořený uživatelský účet. Účty je možné propojit s Google, nebo i jinými platformami. Zároveň je možné vytvářet žákovské účty a přidávat je do tříd. Tuto možnost pro učitele doporučujeme, protože díky tomu budete mít lepší přehled o práci jednotlivých žáků. S programem TinkerCad se pracuje v podstatě celou dobu, proto je dobré mít vše nastavené již od začátku.

### OBSAH SEKCE

1. Základy práce s programem TinkerCad.
2. Modelování částí podvozku.
3. 3D tisk.

## 1. ZÁKLADY PRÁCE S PROGRAMEM TINKERCAD

### Cíl:

Žák zvládá základní práci při tvorbě 3D modelu v TinkerCadu. Žák je tvoří samostatně nové objekty, tyto objekty slučuje a případně rozděljuje. Žák vytváří v objektech díry a používá funkci pravítka.

### Čas:

45 min

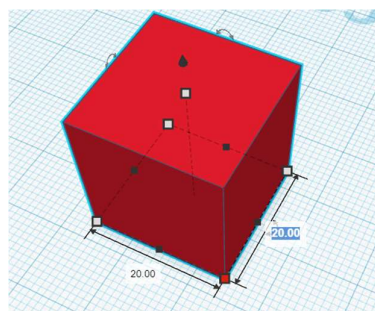
## PRACOVNÍ LIST: ZÁKLADNÍ OVLÁDÁNÍ

### PRÁCE S MYŠÍ

Spolu s dětmi vstupte do prostředí TinkerCadu. Nechte je ať si vyzkouší chování obrazovky při pohybování myši. Žáci v pracovním listu odpoví, co se děje, pokud drží levé tlačítko, pravé tlačítko, nebo kolečko myši a zároveň s ní pohybují.

### NOVÝ TVAR

Nový tvar do pracovní plochy se vkládá jednoduše přetažením požadovaného tvaru z pravého sloupce. Výchozí rozměry kvádrů (respektive kostky) jsou 20x20x20mm. Vysvětlete žákům, že se vše v programu měří v milimetrech a k tomu jsou vytvořené také nákresy v pracovních listech.



Obrázek 1 - objekt s vyznačenou možností přepsání rozměru

### ZMĚNA VELIKOSTI

Základem modelování je skládání jednotlivých tvarů a změna jejich velikosti na požadované rozměry. Změnit velikost je možné buď pomocí bílých čtverců označeného objektu, anebo přímým vepsáním požadované velikosti do pole, které se objeví, pokud na bílé čtverce klikneme.

## PRACOVNÍ LIST: DESKA PODVOZKU

### PRVNÍ KVÁDR

Žáci se poprvé setkávají s technickým nákresem a kótováním. Vysvětlete žákům, co které kóty znamenají a jak tvar vytvořit v TinkerCadu.

### DÍRA PRO KOLO

Podobným způsobem jako v předchozím úkolu nechte žáky vytvořit válec a ukažte jim kde se přepíná režim těleso a režim díra. Rozdíl v těchto režimech je ten, že po sloučení dvou objektů režim těleso rozšíří aktuální tvar, kdežto režim díra vytvoří v tělese díru.

## PRACOVNÍ LIST: PRAVÍTKO

### POČÁTEK SOUŘADNIC

Funkce pravítka nám posune počátek souřadnic do libovolného bodu. To se hodí především v případě, kdy potřebujeme přesně umístit jeden objekt na druhý. Při umísťování pravítka dbejte na jeho přesné umístění. Někdy v důsledku zákrytu ostatními objekty není úplně jednoduché pravítka umístit.

### VÁLEC NA DESKU

S umístěným pravítkem je možné upravovat přesné vzdálenosti mezi počátkem pravítka a objektem. V každém pracovním listu jsou nákresy upraveny tak, aby bylo jednoduché s použitím pravítka objekty umístit.

## PRACOVNÍ LIST: SLOUPKY PRO NOSNÍK I.

### SLOUČENÍ OBJEKTŮ

Důležitá klíčová věc pro práci s Tinkercadem je sloučení objektů. Nejprve je potřeba vybrat několik objektů pomocí klávesy SHIFT + klik. Následně je možné objekty sloučit buď pomocí klávesové zkratky CTRL + G, nebo tlačítkem v nástrojové liště.

---

### DVA KVÁDRY

Žák má za úkol vymodelovat dva kvádry podle nákresu. Pomocí těchto sloupků je vymodelovaný celý podvozek, proto je dobré, aby si žák osvojil jejich tvorbu a určil si, ze kterých tvarů se sloupek skládá.


---

### PRACOVNÍ LIST: SLOUPKY PRO NOSNÍK II.

---

#### SLOUČIT A ZKOPÍROVAT

Umístění objektů přesně na svůj střed je v tomto případě možné jednoduše myší. Pokud chceme být přesnější

tak vyberme objekty, které chceme vůči sobě pohybovat a zvolme nástroj zarovnání . Následně se v okolí obou objektů objeví několik černých teček pomocí nichž je možné objekty vůči sobě zarovnávat. Další důležitou funkcí je kopie. Jakmile má žák sloupek hotový, nemusí již ručně modelovat druhý sloupek, ale může jednoduše hotový sloupek zkopírovat pomocí klávesové zkratky CTRL+C a vložit pomocí CTRL+V.

---

#### UMÍSTĚNÍ SLOUPKŮ

Opět práce s pravítkem probíhá stejným způsobem jako v kapitole Počátek souřadnic

---

### PRACOVNÍ LIST: TŘETÍ BOD

---

#### NÁKRES NOSNÍKU

První složitější díl konstrukce. Nově se žák setkává s kótou značící průměr díry. Postup práce je jednoduchý. Nejprve vytvořte hlavní obdélníkový kvádr a do něj poté vytvořte díry. Díry umístěte pomocí pravítka, které si umístíte do levého dolního rohu obdélníka.

---

#### VĚTŠÍ DÍRY

Důležité. Protože 3D tisk není 100% přesná technologie, tak je potřeba na každé hraně počítat s 0,2mm tlustou vrstvou materiálu navíc. Z toho plyne, že pokud v modelu využíváme nějaké díry, do kterých musíme vkládat jiné části konstrukce, tak díry musí být na každé hraně o 0,2mmvětší, tedy o 0,4mm na každém rozměru. Proto velikost obdélníkových děr je 2,4x4,4mm.

## 2. MODELOVÁNÍ ČÁSTÍ PODVOZKU

### Cíl:

Žák s čím dál menší pomocí modeluje jednotlivé části podvozku. Žák prohlubuje dovednost čtení technických dokumentů. Žák spočítá jednoduché kóty.

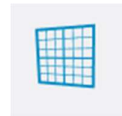
### Čas:

90 min

### PRACOVNÍ LIST: SLOUPKY PRO DESKY

#### NÁKRES

Sloupek se skládá ze 3 kvádrů a dvou válců, pomocí kterých jsou poté vytvořeny díry pro uchycení motorů. Pro správné umístění válců doporučujeme použít nástroj rovina, který nám umožní pracovat v kterékoli jiné rovině. Jednoduše rovinu umístíte na plochu, na kterou chcete válec umístit a do levého dolního rohu umístíte pravítko.



#### UMÍSTĚNÍ

Hotové sloupky opět umístíme na hlavní desku podobně jako u prvních dvou sloupků. Pozor je potřeba sloupky povysunout do výšky 2mm, aby nebyly utopené v podložce.

### PRACOVNÍ LIST: SPODNÍ DESKA

#### VELIKOST DĚR

Veďte žáky k tomu, aby sami přišli na velikost výřezů pro sloupky. Spodní deska musí klesnout až na první schod sloupku a zároveň nezapomeňte, že díry musí být o 0,4mm větší. Výřez spodní desky tedy bude 16,4x2,4mm.

#### VYMODELUJ TO

Modelování spodní desky je jednoduché. Skládá se z 5 kvádrů. Jeden slouží jako deska a 4 slouží k tvorbě děr. Důležité je mít správné rozměry děr a mít je přesně umístěné. To docílíte opět pomocí pravítka.

### PRACOVNÍ LIST: HORNÍ DESKA

Viz pracovní list: Spodní deska. List funguje naprosto shodně s předchozím. Navíc má žák zjistit vnější kóty desky, která bude stejně velká jako deska nad ní.

### PRACOVNÍ LIST: POUZDRO PRO ČIDLA I

Pouzdro se skládá ze 3 kvádrů. Kvádry, které jsou určeny pro tvorbu děr nechte defaultně vysoké tedy 20mm. Proto v nákresu chybí hodnota kóty.

### PRACOVNÍ LIST. POUZDRO PRO ČIDLA II

#### JENGA

Všechny 3 bloky je potřeba naskládat přesně na sebe. Dole musí být umístěný menší z průhledných bloků a nahoře větší z průhledných bloků.

#### ZASUNOUT DOVNITŘ

Horní průhledný blok zasuňte ve svislém směru 5mm do pevného bloku, spodní průhledný blok pak zasuňte 1mm do pevného bloku. Následně všechny bloky slučte dohromady.

### PRACOVNÍ LIST: SLEDOVAČ ČÁRY

#### ROZTEČ

Nechte děti přemýšlet nad tím jaké mají být kde rozměry kót. Výsledek je 9x15mm. 9mm je vnější šířka bloku pro čidlo a 15mm je zadaná rozteč mezi čidly.

---

### VYMODELUJ TO

Rozkopírujte si pouzdra pro čidla tak, abyste jich měli 4 kusy. Mezi tato pouzdra vždy umístěte blok, který definuje rozteč mezi jednotlivými pouzdry. Vznikne jakýsi ucelený blok, který na konci práce umístíme do přední části robota.

---

### PRACOVNÍ LIST: DRÁŽKA PRO ULTRAZVUK

Poslední součástkou, kterou musíme na podvozek umístit je ultrazvukový senzor. Ten se bude nacházet úplně na špičce podvozku.

---

### 3 ZÁKLADNÍ KVÁDRY

Vytvořte a slučte 3 kvádry podle nákresu. Pozor vzadu za svislými kvádry je záměrně nechaných 2mm rezervy. Je to z toho důvodu aby se kontakty čidel sledujících čáru omylem nedotkly desky plošného spoje ultrazvukového senzoru.

---

### DRÁŽKA

Drážku do sloupků vytvoříme pomocí desky, která nám simuluje zasunutý ultrazvukový senzor.

---

### PRACOVNÍ LIST: HOTOVO

Poskládejte právě vymodelované díly na přední část robota a slučte je s již existujícím podvozkem. Kola žákům darujte (mějte je předtištěné) protože jsou poněkud složitější na tvorbu. Šikovnější žáky můžete nechat kola vytvořit. V přílohách jim poskytněte soubor kolo\_robot.svg. Tento soubor obsahuje svg nákres loukoťového kola, který je možný nainportovat do TinkerCadu. Následně je potřeba v kole vytvořit drážku pro gumičku pomocí tvaru Torus.

### 3. 3D TISK

3D tisk není podepřen pracovními listy z toho důvodu, že každá 3D tiskárna se ovládá jiným způsobem a bude se lišit škola od školy. Soubor z TinkerCadu je potřeba vyexportovat a nahrát do programu, který celý model vysliceruje a připraví jej pro 3D tisk. Na samotný tisk je potřeba 5 – 8 hodin času, proto je vhodné plánovat tisknutí tak, abyste jej mohli provést během týdne. Zatímco budete tisknout jednotlivé modely můžete se s žáky pustit do další části a tou je Elektronika v TinkerCadu, ke které není potřeba nic víc než pouze samotný počítač.





### ÚVOD

Žáci za sebou mají první část práce, a to je modelování podvozku. Další část práce se týká zapojení samotné elektroniky. Než se však pustí do práce s reálnými součástkami je vhodné, aby se základům práce žák naučil ve virtuálním prostředí, které taktéž poskytuje program TinkerCad. První část stavby elektroniky je proto věnována práci v online virtuálním prostředí.

#### **Cíl sekce:**

Žák sestaví jednoduchý obvod a spustí simulaci v programu TinkerCad. Žák popíše způsob práce s kontaktním polem. Žák v programu sestaví model obvodu pro řízení robota, spustí jeho simulaci a popíše co program dělá a proč. Žák naprogramuje pomocí bloků jednoduchý program pro testování svého obvodu.

### OBSAH SEKCE

1. Seznámení s kontaktním polem, rozsvícení diody, předřadný odpor
2. Arduino, první program, blikání diodou
3. Digitální vstup, tlačítko, pull-down rezistor
4. Analogový vstup, PWM výstup
5. Simulace robota, diferenční řízení

## 1. SEZNÁMENÍ S KONTAKTNÍM POLEM, ROZSVÍCENÍ DIODY, PŘEDŘADNÝ ODPOR

### Cíl:

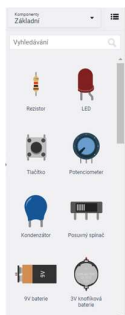
Žák používá další část programu TinkerCad pro tvorbu obvodů. Žák popíše základní ovládání programu. Žák vysvětlí, jak funguje kontaktní pole a k čemu jej používáme. Žák správně připojí do kontaktního pole LED Diodu.

### Čas:

45 min

## PRACOVNÍ LIST: KONTAKTNÍ POLE

### NOVÝ OBVOD



Obrázek 2  
komponenty

Pro úspěšnou práci je potřeba žákům ukázat základní ovládání programu. S programem již žáci pracovali, ale část obvodu má jistá specifika, na která je potřeba upozornit.

#### Komponenty

V pravém sloupci je seznam komponent, které jednoduše přetažením umístíme do plochy, kde s nimi můžeme nadále pracovat.

#### Propojení komponent

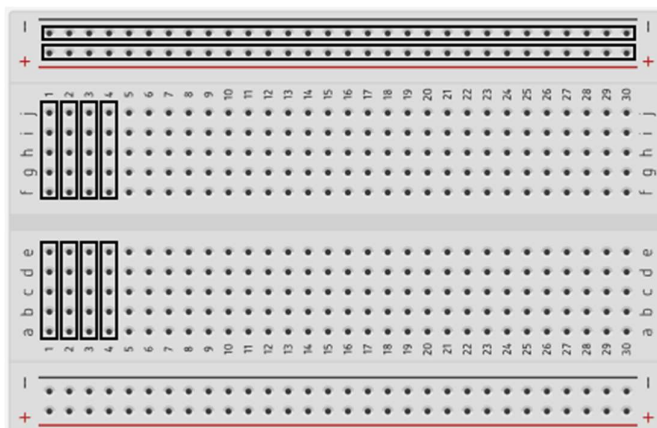
Propojování komponent probíhá prostým kliknutím na vývod součástky a následným kliknutím do druhého místa připojení. Pro přehlednost doporučuji vést všechny spoje pravoúhle a odlišovat je barvami. Typicky se používá červená pro kladný pól napájení a černá pro záporný pól napájení. Další mi barvami je vhodné odlišovat například datové vodiče apod. Barvu vodiče změníme tlačítkem v liště vlevo nahoře.



Obrázek 3 změna barvy  
vodiče

### PROPOJENÍ KONTAKTNÍHO POLE

Kontaktní pole je destička s mnoha dírkami, která slouží pro jednoduché propojování elektrických součástek. Důležité je porozumět jejímu vnitřnímu zapojení. S polem pracujeme vždy tak, aby napájecí dráhy označené červenou a černou čarou byly vždy vodorovně k pozorovateli. Propojeny jsou spolu vždy jednotlivé sloupce (5 dírek nad sebou), které jsou označeny čísly. Dále jsou spolu propojeny napájecí řádky.



Obrázek 4 vnitřní zapojení kontaktního pole

### PŘIDÁNÍ DIODY

Součástky připojujeme vždy tak, aby každý vývod jedné součástky byl ve svém vlastním sloupci.

---

## PRACOVNÍ LIST: ROZSVÍTÍME DIODU

---

### POUŽITÍ LED DIOD

První součástka, se kterou žáky seznámíme se jmenuje LED Dioda.

#### Aktivita:

Zahrajte si s žáky hru, kdy mají ve skupinkách po 3 vymyslet a napsat na papír co nejvíce míst, kde se LED Diody používají. Aktivitu omezte časově cca na 2 minuty.

---

### PLUS A MINUS

LED dioda má dva vývody. Jeden je delší a druhý kratší. Delší vývod se jmenuje ANODA a kratší KATODA. Anodu připojujeme na kladný pól napájení a katodu k zápornému pólu.

#### Mnemotechnická pomůcka:

Pro jednodušší zapamatování, který vývod kam připojujeme se podívejte na slovo **AN**oda, které obsahuje slovo „ANO“. Díky tomu si můžeme zapamatovat, že anoda se připojuje ke kladnému pólu.

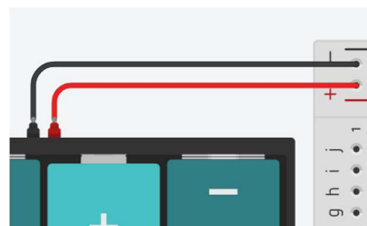
---

### BEZ ENERGIE TO NEPŮJDE

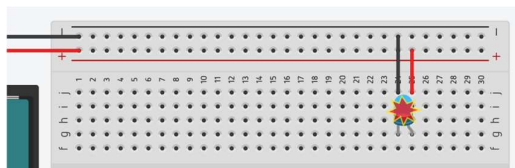
Pro první zapojení použijeme baterii. Pouzdro na baterie mají kladný a záporný pól. Kladný pól se připojuje do plusové řady a záporný do minusové. Doporučuji dodržovat barevné odlišení polarity.

### SVÍTÍ?

Tento úkol je záměrně postavený na chybě. Žákům by měla, po správném splnění úkolu dioda shořet. Vysvětlete žákům na tomto úkolu, že to je důvod, proč používáme simulační programy. Jde o to, že virtuální dioda nás nestojí nic navíc. Může se zdát, že některým žákům se zapojení napoprvé nepovede a budou potřebovat pomoc vyučujícího.



Obrázek 6 připojení napájení do kontaktního pole



Obrázek 5 shořelá dioda v Tinkercadu

---

## PRACOVNÍ LIST: PŘEDŘADNÝ ODPOR

Žáky seznámíme s další součástkou a tou je rezistor. Lidově se také lze setkat s pojmem odpor. Tato součástka slouží pouze k přeměně elektrické energie na energii tepelnou. V předchozím cvičení dioda shořela proto, že jsme do ní pustily větší množství energie, než byla dioda schopná zpracovat. Proto se před diodu vždy zapojuje tzv. předřadný odpor.

---

### HOŘÍ!

#### Aktivita:

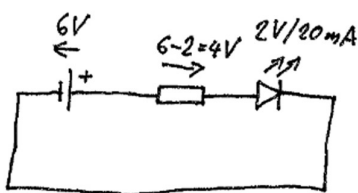
Nechte žáky vyhledat na internetu, jak se jmenuje součástka, která mění elektrickou energii na energii tepelnou.

Vysvětlete žákům jednotku ohm, její značení řeckým písmenem velké  $\Omega$ . Označení veličiny elektrického odporu je R. V dalším kroku se žáci seznámí s výpočtem předřadného odporu.

## VÝPOČET

Tato úloha předpokládá, že se žáci již setkali s jednoduchými vzorci a dosazování hodnot do vzorce. Výpočet a vysvětlení proveďte na tabuli. Můžete využít několik zkušebních příkladů pro zafixování látky.

Nakreslete na tabuli tento jednoduchý obvod.



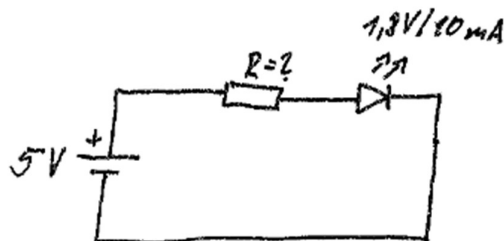
Seznamte žáky s novými značkami pro rezistor, diodu a baterii. Baterie dávají napětí 6V. Dioda ke svému provozu potřebuje cca 2V. V obvodu nám tedy přebývají 4V, které musíme přeměnit na teplo pomocí rezistoru. Vzorec pro výpočet odporu rezistoru vychází z ohmova zákona.

$$R = \frac{U}{I}$$

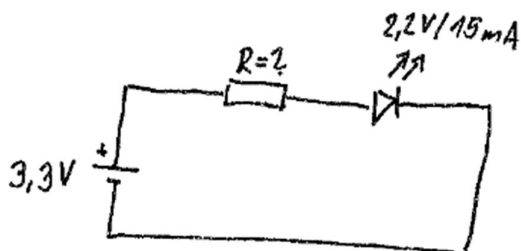
Proud, který prochází diodou a tím i celým obvodem je 20mA. Tento proud nezapomeňte převést do základních jednotek a dosadte do vzorce.

$$R = \frac{4}{0,02} = 200\Omega$$

Další příklady na procvičení:



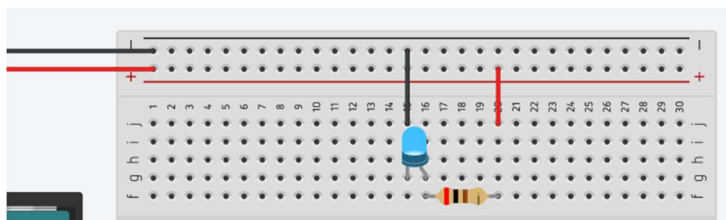
Výsledek:  $R = 320\Omega$



Výsledek:  $R = 73,3\Omega$

## SVÍTÍ, TEĎ UŽ DOOPRAVDY 😊

Žák poprvé zapojí svůj elektrický obvod. Někteří žáci pravděpodobně budou potřebovat poradit, jak zapojit odpor před diodu.



Obrázek 7 výsledné zapojení

## 2. ARDUINO, PRVNÍ PROGRAM, BLIKÁNÍ DIODOU

### Cíl:

Žák připojí napájení kontaktního pole přes piny Arduina. Žák vysvětlí, co je Arduino a co znamená pojem pin. Žák připojí diodu na digitální výstup Arduina a čip naprogramuje tak, aby se dioda rozblíkala. Žák programuje pomocí základních bloků.

### Čas:

45 min

## PRACOVNÍ LIST: ARDUINO

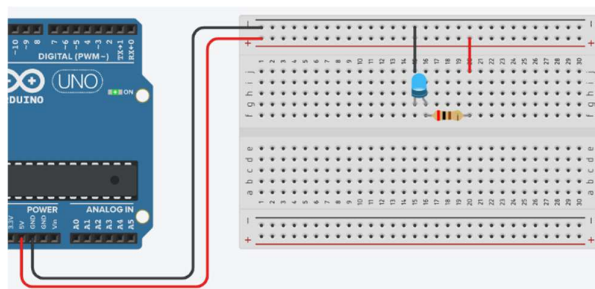
### MOZEK ROBOTY

V následujících cvičeních bude žák pracovat s Arduinem. Ukažte žákům, jak se k pinům připojují vodiče.

Pin je označení pro jeden vývod procesoru. Piny jsou různých druhů. V dalších cvičeních se žák seznámí s dalšími typy pinů.

### NAPÁJENÍ Z ARDUINA

Od teď budeme vše zapojovat pomocí Arduina včetně napájení. Seznamte žáky s napájecími piny. A propojte je s kontaktním polem. Spustěte simulaci ať žáci vidí, že obvod svítí stále jenom jsme vyměnily zdroj napájení. Napájení přes Arduino je možné díky tomu, že Arduino se připojuje pomocí USB kabelu k počítači. Počítač má v USB portu integrovaný napájecí zdroj o velikosti 5V/500mA.



### DIGITÁLNÍ VÝSTUPY

Digitální výstup znamená, že mikroprocesor na svém digitálním pinu může podle programu zapnout (log. 1) nebo vypnout (log. 0) napětí 5V. Žák si může tuto situaci představit například na obyčejném domovním vypínači. Pokud vypínač zapnu, světlo svítí, pokud ho vypnu, světlo je zhasnuté. Stejně tak může Arduino zapnout nebo vypnout svůj výstup.

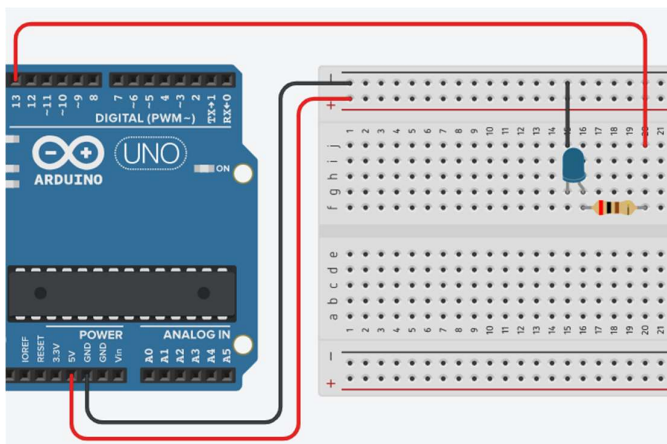
## PRACOVNÍ LIST: HELLO WORLD!

Hello world je označení pro program, který většinou vytvoří každý programátor jako první program ve svém životě. Když se někdo začíná učit nějaký programovací jazyk, jako jedním z prvních programů, který takový uživatel vytvoří je jednoduchý program, který vypíše na obrazovku vzkaz Hello World. V případě elektroniky to bývá právě blikání LED Diodou.

### LEDKU PŘIPOJÍME K ČIPU

Úkolem žáka je smazat propojení mezi rezistorem a plusovým napájením a místo toho propojit rezistor s pinem 13 na Arduinu. Po spuštění simulace by měla LED dioda blikat. Důvod blikání je ten, že uvnitř virtuálního Arduina je již od výrobců TinkerCadu defaultně nastaven program pro blikání diodou (naš Hello World). V dalším kroku se žák s programem seznámí.

Pozor žáky může zmást, že náčrt podle, kterého zapojují obvod není úplný a Arduino má naznačené pouze piny 13 a GND. Vysvětlete žákům, že Arduino je velmi složitý obvod a pro jednoduchost kresby se označují pouze pro zapojení důležité piny. Tímto způsobem budou zakresleny i další schémata.



Obrázek 8 výsledné zapojení k pinu 13

## PROGRAM

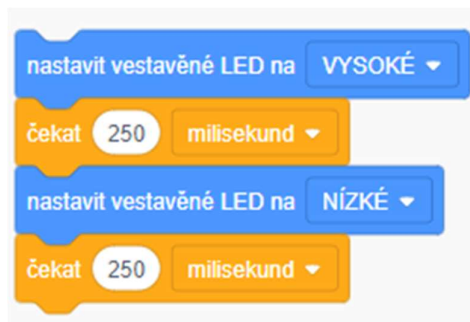
### Základní principy programu Arduina:

Program v mikročipu se nikdy nezastaví a vždy když dojde na konec programu, tak automaticky skočí na začátek a pokračuje.



Arduino UNO má vestavěnou LED diodu, která je zároveň připojená na výstup 13. Upozorněte žáky, že během simulace také bliká dioda přímo na desce Arduina.

Program se sestavuje z jednotlivých bloků. Uvnitř bloků mohou být parametry, které je možné měnit. Vysvětlete žákům, co je **parametr** (v tomto případě jde o číslo, jak dlouho má program čekat a na jakou hodnotu má výstup nastavit).



## ZMĚNA PINU

Jde o cvičení, při kterém si žák osvojí práci s bloky. Žák musí odstranit z programu modré bloky a nahradit je blokem typu: „nastavit kolík # na #####.“ Upozorněte žáky, že hodnota VYSOKÉ odpovídá zapnutému pinu +5V (logická jednička) a hodnota NÍZKÉ odpovídá vypnutému pinu 0V (logická nula).

### 3. DIGITÁLNÍ VSTUP, TLAČÍTKO, PULL-DOWN REZISTOR

#### Cíl:

Žák připojí k Arduino tlačítko, naprogramuje program pro čtení digitálního vstupu. Žák vysvětlí připojení pull-down rezistoru. Žák vysvětlí funkci jednoduché podmínky v programu.

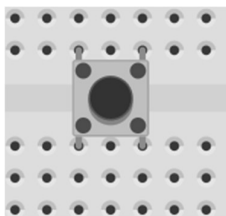
#### Čas:

45 min

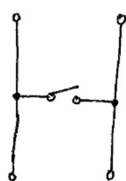
Typickým digitálním vstupem je jednoduché tlačítko. Digitální vstup rozlišuje pouze jedničku (napětí vyšší než 2,5V) a nulu (napětí nižší než 2,5V). Problém s jakýmkoli vstupem procesoru je jeho nedefinovaná hodnota při rozpojení obvodu. Tento stav se řeší pull-down případně pull-up rezistorem.

#### PRACOVNÍ LIST: TLAČÍTKO

##### PŘÍDEJ TLAČÍTKO



Obrázek 10 připojení tlačítka přes žlábek



Obrázek 9 vnitřní zapojení tlačítka

Poprvé v tomto cvičení žák zapojuje součástku, která musí být zapojena přes středový žlábek kontaktního pole. Jedině tak je možné zajistit, aby vývody tlačítka měly každý sloupeček zvlášť.

Vnitřní zapojení tlačítka je na Obrázek 9.

##### DIGITÁLNÍ VSTUP

Pro tlačítko využívám digitální pin 5. Předpokládáme, že žáci mají z předchozího cvičení diodu připojenou na pin 6. Je-li tomu jinak, je možné použít libovolný digitální pin pro vstup, ale pak je nutné i v dalším postupu při programování tuto změnu reflektovat.

##### JEDNIČKA NEBO NULA?

Úkol si dává za cíl dovést žáky myšlenkově k tomu, že při rozpojeném tlačítku je na vstupu procesoru nedefinovaný stav. Vstup není připojen ani k jedničce (+5V) ani k nule (0V). Může se tedy stát, že někdy procesor vyhodnotí hodnotu na pinu 5 jako jedničku a někdy jako nulu. Tomuto stavu říkáme **nedefinovaný stav**. Upozorněte na tento stav žáky. Následující pracovní list vysvětlí, jak tuto situaci vyřešit.

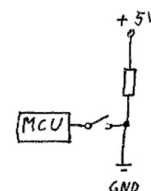
#### PRACOVNÍ LIST: ZNIČ NEDEFINOVANÝ STAV

##### ZKRAT

Vysvětlíte žákům, co je to zkrat a jak se může stát. Upozorněte je, že zkrat je velmi nežádoucí stav elektrického obvodu, který může mít za následek zničení zařízení a součástek. Při větších napětích a proudech může dojít i k újmě na zdraví nebo smrti.

##### PULL-DOWN REZISTOR

Abychom předešli zkratu, vkládáme do obvodu rezistor s velkou hodnotou odporu. Pull-down v překladu znamená stáhnout dolů. Toto zapojení se tak jmenuje proto, že rezistor je připojený mezi vstupním pinem a nulou (GND). Analogicky tomu existuje i pull-up zapojení. To je z hlediska didaktiky složitější, protože při stisknutí tlačítka se na vstup dostane logická nula. V normálním stavu je pak trvale na vstupu logická jednička, což je pro žáky hůře pochopitelné a následné programování je složitější k zamyšlení.



Obrázek 11 pull-up zapojení

#### PRACOVNÍ LIST: JE TLAČÍTKO ZMÁČKNUTÉ NEBO NE?

##### VSTUP DO PROGRAMU

Program uvnitř čipu nezjistí sám od sebe, že na nějakém pinu je nějaká hodnota. I tuto operaci mu musíme explicitně říct. V TinkerCadu najdeme všechny funkce pro čtení vstupů v sekci vstupy. Jsou označeny fialovou barvou.

---

### ZAPAMATUJ SI

Hodnotu, kterou čip vyčte je potřeba v programu uložit v proměnné. Vysvětlete žákům, co jsou proměnné a jak si je vytvoří. Pozor program je vyvinutý v USA a nepodporuje proto diakritiku. Názvy proměnných se musí vytvořit bez diakritiky.

---

### POKUD ZMÁČKNU TLAČÍTKO TAK

Blok podmínky je základním stavebním kamenem každého programu. Pokud je splněna nějaká podmínka, tak program provede nějaký kód, který je uvnitř podmínky uveden. Vysvětlete žákům, že kód, který má být proveden se vkládá dovnitř do bloku s podmínkou. Do místa mezi slovy „pokud“ a „pak“ se vkládá operátor porovnání, který pokud je vyhodnocen kladně, tak program provede kód uvnitř podmínky.

#### **Aktivita:**

Zeptejte se žáků, co by asi tak mohlo uvnitř podmínky být. Cílem je, aby žáci myšlenkově dospěli k tomu, že pokud zmáčknu tlačítko, tak se rozsvítí LED dioda.

---

### OPERÁTOR

V souvislosti s předchozí aktivitou připomeňte žákům porovnávací operátory, které již znají z matematiky (< je menší, > je větší, = rovná se), případně můžete zmínit je menší nebo rovno a je větší nebo rovno.

---

### PRACOVNÍ LIST: OVLÁDEJ DIODU

Provedte žáky sestavením prvního programu.

---

### POKUD MAČKÁM, TY SVÍTÍŠ

Snažte se o neustálé vysvětlování programu českými větami. Pokud je tlačítko zmáčkuté, a tedy se rovná jedničce, tak nastavíme kolík, na kterém je připojená dioda na vysokou hodnotu, tedy jedničku a dioda tak bude svítit.

#### **Aktivita:**

Před pokračováním na další úkol. Nechte žáky, aby slovy sestavily program opačný. Tedy že pokud stisknu tlačítko, dioda zhasne.

---

### POKUD NEMAČKÁM, TY SVÍTÍŠ

Jde o obdobné cvičení, žák má upravit již existující program dojde tím k zafixování a lepšímu pochopení principů programování.



## 4. ANALGOVÝ VSTUP, PWM VÝSTUP

### Cíl:

Žák popíše rozdíl mezi digitálním a analogovým vstupem. Žák zapojí a naprogramuje jednoduchý obvod obsahující fotorezistor. Žák popíše, jak fotorezistor funguje. Žák popíše, jak funguje pwm. Žák zapojí jednoduchý obvod s využitím pwm výstupu.

### Čas:

45 min

### PRACOVNÍ LIST: SVĚTLO

Žák se pomocí tohoto listu seznámí s novou součástí a tou je fotorezistor. Objasněte žákům, proč se tato součástka připojuje k analogovému pinu.

V principu jde o to, že světlo, na rozdíl od tlačítka může nabývat různých intenzit. Lidově těmto intenzitám říkáme přitímní, ostré světlo, tma a podobně. Fotorezistor na tyto změny reaguje a mění svůj odpor v závislosti na intenzitě světla. Rozpoznáváme tedy mnohem více hodnot než pouhé 2, a proto je nutné fotorezistor připojit k analogovému vstupu, který je schopen rozeznávat až 1024 hodnot.

### FOTOREZISTOR

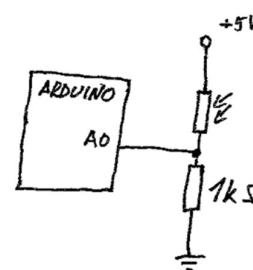
Objasněte žákům, že jde o součástku, která reaguje na světlo. V závislosti na tom, kolik na ni dopadá světla, tak mění svůj odpor. Slovo foto tedy značí, že součástka je ve své podstatě rezistor, ale reaguje na světlo.

### ZNOVU PULL-DOWN

Stejným způsobem, jakým jsme připojovali tlačítko je potřeba připojit také fotorezistor. Existují pro toto zapojení úplně stejné důvody. V okamžiku, kdy na fotorezistor nebude dopadat žádné světlo, tak bude uzavřený a vstupní analogový pin se ocitne v nedefinovaném stavu.

### ČÍST ANALGOVÝ KOLÍK

Stejným způsobem jako jsme četli digitální kolík s připojeným tlačítkem, tak se pracuje i s analogovými kolíky. Žákům neustále připomínejte podobnost práce s digitální částí.



Obrázek 12 Zapojení fotorezistoru

### PRACOVNÍ LIST: KOLIK JE TU SVĚTLA?

### SÉRIOVÝ MONITOR

Sériový monitor je způsob, kterým s námi může procesor komunikovat. Umožňuje vytisknout na obrazovku počítače hodnoty proměnných. V tomto pracovním listu budeme zjišťovat jakých hodnot nabývá proměnná fotorezistor v okamžiku, kdy je světlo nebo tma.

Vysvětlete žákům, že na skutečném Arduinu je také možné použít sériový monitor. Arduino pak komunikuje s počítačem pomocí USB kabelu.

### 0 AŽ 1024

Spusťte simulaci a klikněte na součástku fotorezistor. Objeví se nad ní posuvník a tím simulujete množství dopadajícího světla na součástku. Při otevřeném sériovém monitoru se vám budou vypisovat ve spodní části obrazovky hodnoty mezi 0 a 1023. Zapište si s žáky hodnotu, když je fotorezistor v úplně tmě a když je v největším světle.

### Pro zvědavé žáky:

Proč hodnoty od 0 do 1023. Vysvětlete žákům, že to souvisí s bitovým rozlišením vstupního AD převodníku. U Arduina je tento převodník 10bitový a ten dovede zaznamenat hodnoty od 0 do 1023.  $2^{10} = 1024$  hodnot.

### KDYŽ JE TMA TAK ROZSVÍŤ

Cílem je naprogramovat program, který při setmění rozsvítí diodu a při osvětlení se světlo zhasne. Na podobném principu fungují ozdobné zahradní lampičky. Podívejte se spolu s žáky na hodnoty z předchozího cvičení a najděte

+ - středovou hodnotu mezi tmou a světlem. Většinou by měla být okolo 400 – 600. Toto je naše **prahová hodnota**, při které program rozliší světlo nebo tmu.

---

## PRACOVNÍ LIST: STMÍVÁ SE

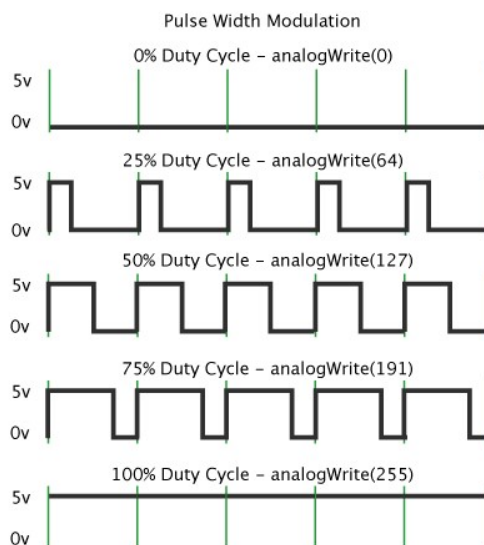
### PWM VÝSTUPY

#### Motivace (proč to děláme):

Digitální čip dovede na výstup dodat pouze jedničku nebo nulu. Jak ale pomocí digitálního čipu řídit například rychlost motoru, což budeme potřebovat pro úspěšné dokončení robota.

Žáci se seznámí s funkcí PWM. Pwm je digitální způsob simulace analogového signálu. Čip si rozdělí určitou časovou periodu na 256 stejných dílů. Potom uvede výstup do logické hodnoty 1 pouze po určitý počet těchto dílů. Čím déle bude výstup v logické 1, tím více energie se dostane do připojeného zařízení (LEDka nebo i motor).

Ne všechny výstupy Arduina tuto možnost PWM mají. Výstupy, které touto funkcí disponují jsou označeny vlnovkou.



Obrázek 13 Pulzní šířková modulace

### PŘIPOJ DO PWM DIODU

Pokud si ji žáci neodpojili, dioda by po předchozích cvičeních měla být správně připojena k pinu 6, který funkcí PWM má. Pro jistotu si tuto skutečnost žáci zkontrolují.

### 0 AŽ 256

Žáci mají smazat dosavadní program kromě prvního růžového bloku. Pod tento blok zapojte modrý blok **nastavit kolík # na ###**. Tento blok umožňuje pouze digitálním PWM výstupům přiřadit hodnotu od 0 do 255. Jde o příkaz, který v jazyce Wiring znamená `analogWrite(###);`

### O KOLIK MUSÍME HODNOTU ZMENŠIT?

Protože na vstupu fotorezistoru máme hodnotu od 0 do 1024, pak tuto hodnotu nemůžeme prostě vzít a nastavit ji na výstup. Hodnotu je nutno vydělit 4, aby se nám vešla do osmibitového limitu PWM výstupu.

---

## PRACOVNÍ LIST: INTENZITA SVĚTLA

### MATEMATIKA

Je aplikací předchozího výpočtu. Žáci zjistili, že je potřeba hodnotu vydělit 4 a nyní musí pro tuto funkci najít odpovídající blok v TinkerCadu. Upozorněte žáky na tvar bloků. Bloky, které reprezentují nějakou hodnotu mají oválný tvar.

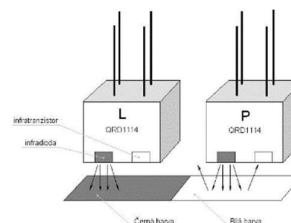
### HOĎ TO NA VÝSTUP

Celý výpočet z předchozího cvičení má oválný tvar. Zeptejte se žáků, co jsme pomocí tohoto oválného bloku spočítali za číslo. Odpověď by měla znít, že jde o hodnotu na fotorezistoru, kterou jsme vydělili čtyřmi, abychom ji mohli použít na pwm výstup.

### JAK TO POUŽÍT U ROBOTA?

Jde o návrat k našemu robotovi. Žáci již poměrně dlouho se seznamují se základy elektroniky a může jim unikat pravý cíl.

Můžete žáky pomalu uvádět do principů sledování černé čáry. Princip spočívá v rozdílném odražení světla černou a bílou barvou. Snímač světla tak má jinou



Obrázek 14 Princip odrazu světla

hodnotu, pokud se nachází nad černou čarou a jinou, pokud je nad bílým povrchem, který odráží světlo velmi dobře. Díky tomuto principu lze rozeznat, kde se robot nachází.

## 5. SIMULACE ROBOTA, DIFERENČNÍ ŘÍZENÍ

### Cíl:

Žák vytvoří obvod, který mu pomůže simulovat chování robota. Žák naprogramuje jednoduchý program pro diferenční řízení robota. Žák vysvětlí, jak funguje diferenční řízení.

### Čas:

45 min

Žák sestrojí obvod, který budeme používat pro naprogramování a simulaci virtuálního robota. Reální robot má dvě kola a dva senzory pro sledování čáry. Virtuální model postrádá ultrazvukový senzor.

Pozor v zapojení je silné zjednodušení. Motory nesmí být připojeny přímo k procesoru, protože mají velký proudový odběr a Arduino není schopné tuto dávku proudu poskytnout. Pro připojení motorů budeme používat tzv. H můstek.

Zapojení senzorů je také zjednodušené, protože skutečná čidla se skládají ze dvou součástek. Z infračervené diody a fototranzistoru. Pro simulační program a pro jednoduchost si však vystačíme s tímto zjednodušením.

### PRACOVNÍ LIST: VIRTUÁLNÍ ROBOT (HARDWARE)

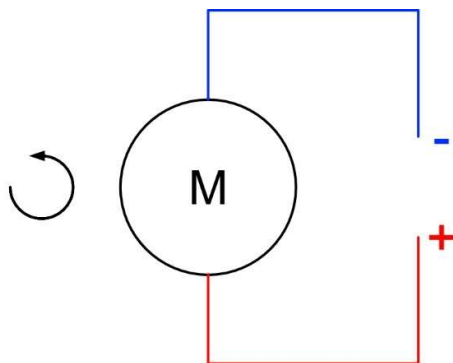
#### DVĚ ČIDLA

Zapojení využívá předchozích osvojených znalostí. Jedná se o zapojení pull-down rezistorů. Je dobré si čidla pojmenovat, aby bylo zřejmé, které čidlo je levé a které pravé. To stejné platí i pro motory v dalším příkladu. Pro lepší vysvětlení se vraťte k předchozím kartám, kde je vysvětlen princip zapojení pull-down rezistoru.

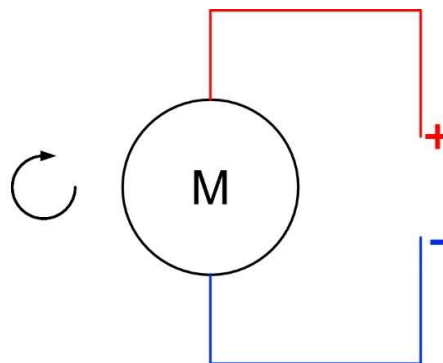
#### DVA MOTORY

Zapojení využívá připojení motorů k pwm výstupům. Opět je možný návrat k předchozím kartám pro lepší vysvětlení.

Vysvětlíte žákům, jak se ovládá směr a rychlost otáčení motorů.



Obrázek 15 Otáčení motoru doleva



Obrázek 16 Otáčení motoru doprava

Princip spočívá v polaritě zapojení. Zároveň je možné řídit rychlost otáčení motoru, tato rychlost se řídí pomocí pwm. Jeden výstup bude vždy v nule. Druhému výstupu pro motor nastavíme požadovanou rychlost v rozmezí 0 – 255.

---

## PRACOVNÍ LIST: VIRTUÁLNÍ ROBOT (ŘÍZENÍ)

---

### VLEVO NEBO VPRAVO?

**Aktivita:**

Rozdělte děti do dvojic. Dvojice se vždy chytí za ruce. Žáci ve dvojici představují levý a pravý motor robota. Pokud chce dvojice zatočit vpravo, jeden z dvojice musí stát a druhý jít vpřed. Nechte žáky, ať sami zjistí, který to je.

Předchozí aktivita by měla žáky dovést k tomu, jak funguje diferenční řízení. Když jedno kolo stojí a druhé jede. Robot se začne otáčet. Následně žáci doplní do obrázku, které čidlo snímá světlo a které tmu. Dále žáci určí, které kolo se bude otáčet vpřed, a které bude stát, aby se robot stočil podél čáry.

---

## PRACOVNÍ LIST: VIRTUÁLNÍ ROBOT (DIAGRAM)

---

### VÝVOJOVÝ DIAGRAM

Vysvětlete žákům, jak fungují vývojové diagramy. Postupně s žáky projděte na tabuli vývojový diagram a provedte je logickým uvažováním nad tím, co se právě s robotem děje, a jak musí program zareagovat na změny čidel. Upozorněte žáky na to, že program v Arduinu nemá žádný konec a musí se neustále opakovat dokola, to je znázorněno šipkou směřující zpátky k prvnímu bloku.

---

## PRACOVNÍ LIST: VIRTUÁLNÍ ROBOT (SOFTWARE)

---

### PROGRAM

Vysvětlete žákům, že podmínky je možné také spojovat pomocí „a“ nebo „nebo“. Rozdíl v těchto blocích je, že v prvním případě musí platit obě podmínky zároveň. Druhá možnost vyjadřuje platnost jedné nebo druhé podmínky.

V poslední podmínce jsou spojeny dvě podmínky dohromady. Zároveň musí platit, že levý i pravý senzor jsou na bílém povrchu, potom má robot jet na plný výkon vpřed.

Nakonec spusťte s žáky simulaci a vyzkoušejte si, zda se motory točí tak, jak jsme očekávali.



Obrázek 17 spojování podmínek

## SKUTEČNÉ ZAPOJENÍ

### Cíl sekce:

Žák sestaví robota, který bude schopen sledovat černou čáru.

### ÚVOD

Žáci se seznámili s absolutními základy práce s elektronickými součástkami. Dokonce si vyzkoušeli i jednoduchou simulaci výsledného robota. Nyní se pustí do práce se skutečnými součástkami. Tato část práce staví na vědomostech z části předchozí. Pozor některá zapojení byla v předchozí části zjednodušena a s reálnými součástkami by nefungovala. Na tato zjednodušení vás metodické listy upozorní. Na konci této sekce bude sestaven základní robot, který bude schopen jezdit po čáře a zastaví se před překážkou. Touto sekcí zároveň končí nejjednodušší základní část práce. Následují metodické listy, které rozšiřují dosavadní zkušenosti a kladou rozvíjející otázky.

### POMŮCKY

- Pájecí hrot
- Štípací kleště
- Cín
- Propojovací vodiče
- Kontaktní pole
- Vytištěný podvozek robota
- 

### OBSAH SEKCE

1. Příprava a zapojení motorů
2. Zapojení senzorů čáry
3. Naprogramování robota

## 1. PŘÍPRAVA A ZAPOJENÍ MOTORŮ

### Cíl:

Žák připájí vodiče k motorům a osadí motory na podvozek robota. Žák otestuje správnost svého zapojení pomocí předpřipraveného programu. Žák nahraje program do Arduina.

### Čas:

45 min

## PRACOVNÍ LIST: ZÁKLADY

### NAPÁJENÍ A ARDUINO

Z počátku může být matoucí, že Arduino má jinou podobu než to, které jsme používali ve virtuálním zapojení. Vysvětlíte žákům, že Arduino, se kterým pracují je Arduino Nano a jeho vývody jsou úplně ty stejné jako u Arduina Uno, které jsme používali v TinkerCadu. Proto můžeme téměř 1:1 převzít všechna zapojení z TinkerCadu a zapojit je na kontaktním poli.

## PRACOVNÍ LIST: MOTORY

### Pozor!

V této části práce se pracuje s pájecím hrotem a roztaveným horkým címem a ostrým zalamovacím nožem.

### PŘÍPRAVA VODIČŮ

Každý žák si vezme dvě dvojice vodičů a na jednom konci jim odstříhne pomocí štípacích kleští připevněné konektory. Následně je potřeba odříznout cca poslední centimetr gumové izolace a zanechat pouze měděné konce.

### PÁJENÍ VODIČŮ

Žákům nejprve předvedte, jak se vodiče připájí k motorům. Upozorněte žáky na nebezpečí popálení.

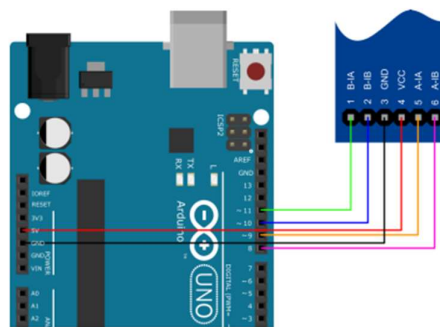
## PRACOVNÍ LIST: H-MŮSTEK

Ve virtuálním prostředí jsme zapojovali motory napřímo k Arduinu. V reálném světě toto není možné, protože Arduino není schopné poskytnout dostatečný výkon pro ovládání motorů. Proto se k zapojení používá H-můstek.

### VÝVODY

Žák má za úkol popsat jednotlivé vývody H-můstku. Vysvětlíte žákům jejich funkci. Velké výstupní svorky s názvem „Motor A“ a „Motor B“ slouží pro připojení samotných motorů. Do jedné svorky připojíme pravý a do druhé levý motor.

Na druhé straně obvodu je 6 vstupních sverek. Uprostřed jsou dva piny pro připojení napájení. Vlevo a vpravo od těchto dvou pinů jsou řídicí svorky, které se připojují přímo na čip.



### PŘIPOJ NAPÁJENÍ A MOTORY

Zapojte s žáky napájení. Napájení připojte přímo na kontaktní pole do řádků + a -. Do velkých výstupních sverek připojte motory. Polarita je v tuto chvíli bezpředmětná. V pozdější fázi se pravděpodobně stane, že některým žákům se bude motor točit obráceně, než by měl, ale to se jednoduše napraví přepojením vývodů u Arduina.

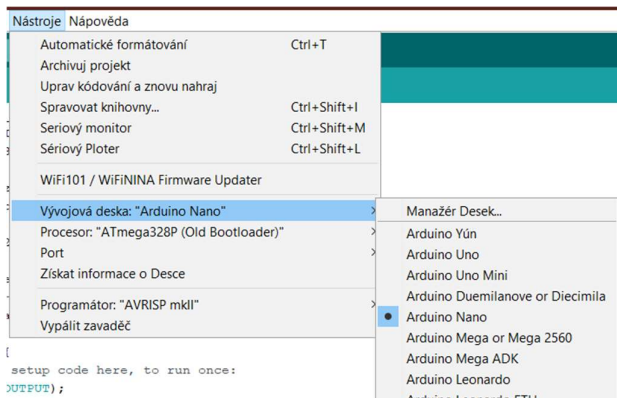
## PRACOVNÍ LIST: PŘIPOJENÍ A TEST

Žáky čeká první test Arduina na reálném zapojení. Předtím než s žáky budete nahrávat program zkontrolujte jejich zapojení.

### ZAPOJ ŘÍZENÍ

Pro zjednodušení budoucí práce zajistěte, aby všichni žáci připojily levý motor k pinům 5 a 6 a pravý motor k pinům 9 a 10.

## TEST MOTORŮ



Obrázek 18 - Arduino IDE nastavení

dělá pouze to, že roztočí oba motory pět vteřin vpřed a následně pět vteřin vzad.

## OPRAVA OTÁČENÍ

Pravděpodobně se některým žákům stane, že se motory točí obráceně, než by měly. Potom jednoduše prohodte mezi sebou piny 5 a 6 nebo 9 a 10 podle toho, který motor se otáčí obráceně.

Vysvětlete žákům, jak se nahrává program do Arduina. Připojte Arduino USB kabelem k počítači. Otevřete si s nimi Arduino IDE a proveďte je nastavením v menu nástroje -> vývojová deska Arduino Nano, procesor ATmega328P (Old Bootloader). Nový Bootloader použijte pouze v případě, že používáte originál Arduino Nano a ne jejich čínské kopie. Port zvolte aktuálně používaný port. Nejčastěji metodou pokus omyl zjistíte, který to je. V případě, že program nelze nahrát nejčastěji je chyba právě v těchto nastaveních.

V příložených souborech naleznete soubor motory\_test.ino. Ten si otevřete v programu Arduino IDE a nahrajte jej do procesoru. Program



## 2. ZAPOJENÍ SENZORŮ ČÁRY

### Cíl:

Žák pojmenuje jednotlivé části součástky QRD1114. Žák vypočítá předřadný odpor před diodu. Žák vysvětlí co je infračervené světlo. Žák zapojí součástku QRD1114 podle nákresu.

### Čas:

45 min

### PRACOVNÍ LIST: SENZOR ČÁRY I.

#### ROZLOŽ TO NA ČÁSTI

Žáci se seznamují s novou součástkou QRD 1114. Tato součástka jsou ve skutečnosti 2 součástky v jednom. Průhledná součástka je obyčejná LED dioda, která ale svítí v infračerveném spektru. Druhá součástka je fototranzistor, který snímá odražené světlo.

#### CO KAM ZAPOJIT?

Upozorněte žáky, že obě tyto součástky mají jednu nožičku delší a jednu kratší. Stejně jako u LED diody se i u fototranzistoru připojuje delší nožička směrem k plus a kratší směrem k minus. Fototranzistor jsme při simulacích v TinkerCadu nahrazovali fotorezistorem.

#### PŘEDŘADNÝ ODPOR DIODY

Stejně jako při zkušebním zapojení v TinkerCadu i zde musíme spočítat předřadný odpor před diodu. Viz kapitola Pracovní list: Předřadný odpor.

### PRACOVNÍ LIST: SENZOR ČÁRY II.

#### UMÍSTI SENZORY

Pro nejjednodušší zapojení a lepší názornost při dalším programování robota využijte s žáky pouze dvě vnější pozice pro senzory čáry. Vnitřní pozice se použijí v pokročilé části práce.

#### ZAPOJENÍ LED DIODY

Součástky jsou již umístěné v podvozku a žáci mají za úkol zapojit LED diodu. To je ta průhledná ze dvou součástek QRD 1114. Proto nejdříve určete, které dva vývody patří k diodě a ty poté zapojte s žáky spolu s předřadným rezistorem do kontaktního pole. K propojení senzoru a kontaktního pole použijte propojovací kabely s koncovkami samice – samec.

#### ZAPOJENÍ FOTOTRANZISTORU

Druhá část QRD 1114 je fototranzistor, ten je potřeba připojit podobně jako fotorezistor v TinkerCadu s využitím pull-down rezistoru.

### PRACOVNÍ LIST: TEST SENZORŮ

#### NEVIDITELNÉ SVĚTLO

Vysvětlete žákům, že infračervená dioda sice svítí, ale svítí takovým světlem, které není schopné lidské oko zachytit. Avšak čočka mobilního telefonu toto světlo zachycuje a na displeji jej zobrazuje fialovo bílou barvou. Pozor existují už i telefony, které mají infračervené spektrum odfiltrované a nemusí se proto na všech telefonech správně zobrazovat.

Pokud čidla nesvítí, pak je potřeba najít chybu v zapojení.

#### TEST

Program test\_cidla.ino je nastaven tak, aby při zakrytí pravého čidla se točil pravý motor a při zakrytí levého čidla se točil levý motor. Pokud se toto děje křížem, pak je potřeba prohodit zapojení A0 s A1.

### 3. NAPROGRAMOVÁNÍ ROBOTY (KAČENKA)

#### Cíl:

Žák za pomoci sériového monitoru zjistí hodnoty čidla nad černým a bílým povrchem. Žák z těchto hodnot zjistí prahovou hodnotu a vysvětlí její význam. Žák aplikuje zjištěnou hodnotu v programu. Žák stáhne program z TinkerCadu a nahraje jej do Arduina.

#### Čas:

20 min

---

#### PRACOVNÍ LIST: PRAHOVÁ HODNOTA

---

##### MĚŘENÍ SVĚTLA

Podobně jako jsme zjišťovali hodnotu na fotorezistoru v předchozí sekci, se pracuje i se sériovou linkou na reálném Arduinu. Po nahrání programu do Arduina ukažte žákům, kde si mohou zobrazit sériový monitor.

Na sériovém monitoru se začne každých 0,5s zobrazovat informace o číselné hodnotě, kterou Arduino registruje na pravém a levém čidlu. Střídatě pokládejte robota na čáru tak, aby levé a pravé čidlo leželo na černém a bílém podkladu. Zapište hodnoty do tabulky. Měření nemusí být nějak obzvláště přesné, pohybujte se s klidem s přesností +/- 20.

---

##### PRAHOVÁ HODNOTA

Vysvětlíte žákům, proč hledáme prahovou hodnotu. Tato hodnota bude referenční hodnotou, při které se program robota rozhodne, zda je senzor na černém nebo bílém podkladu. Tato hodnota by měla být přibližně v polovině mezi maximální a minimální hodnotou.

Je samozřejmě možné, mít prahové hodnoty zvlášť pro pravý a levý senzor, ale pro jednoduchost doporučujeme tyto hodnoty sjednotit. Obvykle tato hodnota vychází přibližně mezi 300 až 500. Velmi však záleží na aktuálních světelných podmínkách a samotné konstrukci robota, respektive výšce čidla nad povrchem.

---

##### ÚPRAVA PROGRAMU

Znovu s žáky otevřete TinkerCad a program, který jsme programovali v předchozí části. Tento program bude bez problému funkční i na reálném zapojení, pokud byly dodrženy jednotlivé vstupy a výstupy Arduina. V programu pouze změňte všechny prahové hodnoty. Jsou zde celkem 4 a nachází se v podmínkách programu.

V případě, že bude robot špatně reagovat na čáru, pak je vhodné ladit právě tyto prahové hodnoty.

---

#### PRACOVNÍ LIST: PROGRAM KAČENKA

---

##### STÁHNOUT KÓD

Ukažte žákům ikonu, pomocí které mohou kód stáhnout. Žákům se z webového prohlížeče stáhne soubor s příponou .ino. Tento soubor otevřete v programu Arduino IDE a nahrajte jej do Arduina stejným způsobem jako předchozí testovací programy.

---

##### POLOŽ ROBOT NA ČÁRU A SPUSŤ

Závěrečná dlouho očekávaná fáze tvorby robota. Jeho první spuštění na dráhu. S velkou pravděpodobností se u mnoha robotů vyskytnou různé problémy, pak je potřeba najít jejich důvod.

---

#### PRACOVNÍ LIST: ROBOT NEJEDE

Nejde o pracovní list v pravém slova smyslu. Je zde pouze jako rada pro případ, že se robot nechová podle našich představ. Zkontrolujte s dětmi dostatečné nabití powerbanky

## SKUTEČNÉ ZAPOJENÍ – SENZOR PŘEKÁŽEK

### **Cíl sekce:**

Žák k existujícímu robotovi přidá ultrazvukový senzor a upraví program tak, aby robot zastavil před překážkou.

### ÚVOD

Robot jezdí. Schválně žákům položte na čáru nějakou překážku, například penál, nebo krabici. Roboti bezmyšlenkovitě narazí do překážky. Tato sekce rozšiřuje robota o další senzor, kterým je ultrazvukový senzor. Pomocí tohoto senzoru bude robot schopen rozeznat, že se před ním překážka nachází a stihne včas zastavit.

### OBSAH SEKCE

#### 1. Zapojení a naprogramování senzoru

## 1. ZAPOJENÍ A NAPROGRAMOVÁNÍ SENZORU

### Cíl:

Žák popíše jednotlivé vývody ultrazvukového senzoru. Žák vysvětlí, na jakém principu senzor funguje a uvede příklady dalšího využití. Žák zapojí ultrazvukový senzor a přeprogramuje robota tak, aby se vždy bezpečně zastavil před překážkou.

### Čas:

20 min

### PRACOVNÍ LIST: ULTRAZVUKOVÝ SENZOR

Seznamte žáky s novou součástí, a to je ultrazvukový senzor. Vysvětlete, jak součástka funguje. Můžete uvést další příklady, kde se principu ultrazvuku užívá například radar na silnici, nebo echolokace netopýrů v prostoru.

#### VÝVODY

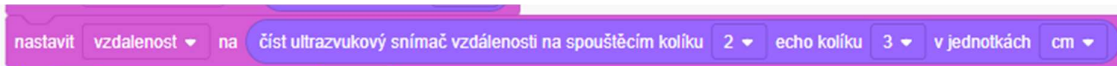
Nechte žáky, aby si součástku prohlédli a určili, které vývody součástka obsahuje.

#### ZAPOJENÍ

Ultrazvukový senzor je velmi jednoduché zapojit. Není potřeba dalších součástek. Zajistěte, aby žáci zapojili vývody správně k Arduino. Ulehčíte si práci s programováním.

#### NOVÁ PROMĚNNÁ

Stejně jako každý vstup do procesoru je řešen proměnnou, tak i ultrazvukový senzor nám vstupuje do programu jako proměnná. Tentokrát si ji pojmenujeme vzdálenost, protože se jedná o vzdálenost k nejbližší překážce. Pro práci s ultrazvukovým senzorem TinkerCad přímo poskytuje funkci. V této funkci je potřeba správně nastavit vstupní piny a jednotky, ve kterých se výsledná vzdálenost měří.



### PRACOVNÍ LIST: POZOR PŘEKÁŽKA

Celý program bude potřeba trochu upravit. Do programu budou přidány dvě nové podmínky. Dovnitř těchto podmínek bude přesunut celý současný kód.

#### POKUD NEVIDÍŠ PŘEKÁŽKU

Někde bokem mimo hlavní program si vytvořte nový podmínkový blok pokud je vzdálenost od překážky větší než 10 cm.

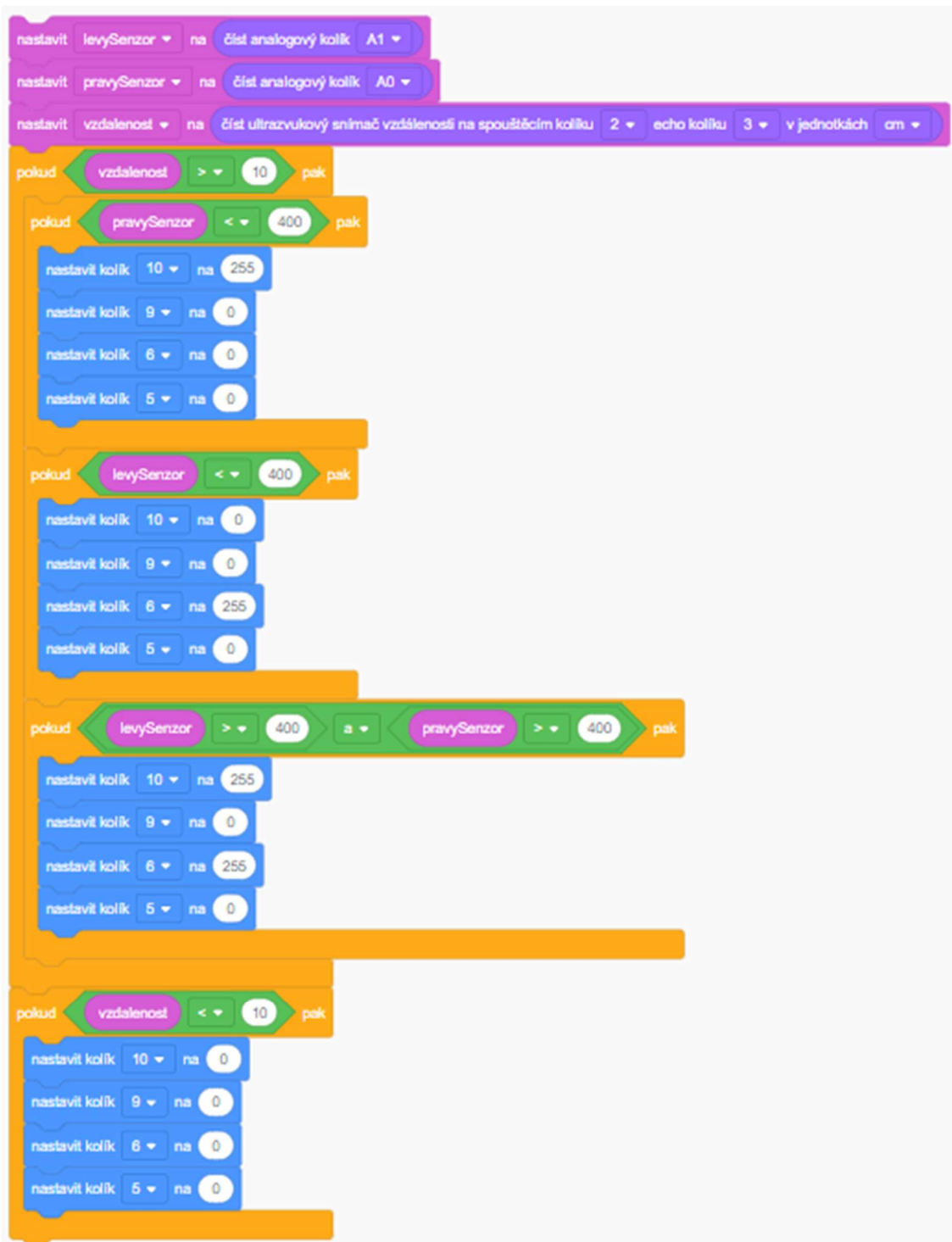
#### TAK JEĎ NORMÁLNĚ

V tomto případě nebude robot zastavovat a program se bude chovat normálně tak, jak se choval do teď. Do této podmínky tedy umístíte veškeré dosud vytvořené podmínky, které řídí robota.

#### POKUD VIDÍŠ PŘEKÁŽKU, ZASTAV

V opačném případě, tedy když je překážka blíž než 10 cm tak je potřeba robota zastavit. Tedy na všech výstupech, kde jsou zapojené motory nastavit nulu.

Kompletní kód je zobrazen zde:



Pro tento program je také možné použít blok „pokud ... jinak ...“. Do tohoto bloku pak stačí vložit pouze jednu podmínku a část bloku jinak se provede v případě, že první podmínka neplatí.

## SKUTEČNÉ ZAPOJENÍ – 4 ČIDLA A ENKODÉR

### Cíl sekce:

Žák prostuduje další možnosti vylepšeného programování robota. Žák sdělí svůj názor a popíše svou motivaci k další práci s robotem. Žák samostatně pracuje a vymýšlí nové algoritmy pro řízení robota.

### ÚVOD

Základní konstrukci robota včetně jeho naprogramování máme za sebou. Poslední dvě sekce se již nevěnují problematice do hloubky, jsou zde pouze jako rozšiřující materiál, který má za cíl žáka nasměrovat k dalším možnostem práce s robotem. Je zřejmé, že do dalších částí se pustí pouze velmi motivovaní a technicky zdatnější žáci. Proto jsou i pracovní listy označeny žlutou a červenou barvou.

---

#### PRACOVNÍ LIST: ZAPOJENÍ DALŠÍCH ČIDEL

Seznamte žáky s dalšími možnostmi práce na robotovi. Prostudujte spolu s nimi příložený vývojový diagram v pracovních listech. Zhodnoťte dosavadní práci a zjistěte jejich výslednou motivovanost k další práci. Žáci, kteří jsou motivovaní pro další práci mohou samostatně vymyslet další způsoby řízení robota.

---

#### PRACOVNÍ LIST: ENKODÉR

Žáci použijí další 4 čidla navíc. Každá dvě čidla směřují směrem k loukotím kola. Princip fungování enkodéru je popsán na této webové stránce, a to včetně kódu pro Arduino:

<https://howtomechatronics.com/tutorials/arduino/rotary-encoder-works-use-arduino/>

Cílem je spíše žáky seznámit s principem fungování než se samotnou konstrukcí.

---

#### PRACOVNÍ LIST: VYPADLÉ DRÁTKY

Pravděpodobně během konstrukce robota se stalo, že se vyskytl nějaký problém a důvodem tohoto problému byl vypadlý drátek z kontaktního pole. Tomu lze zabránit využitím pájivého univerzálního pole. Pracuje se zde však s cínem a horkým hrotem. Proto doporučujeme tuto možnost využít pro starší žáky. Na odborných školách zaměřených na elektrotechniku je dokonce pájení jako jedna z žádaných dovedností.

